「初学者が学習時に躓く知識集」

Agenda

1. 量子力学関連
2. 数学関連
3. Qiskit関連

1. 量子力学関連

a) 量子ビットとは

古典コンピュータのビット（**古典ビット**）に対応する状態の事を言う。

古典コンピュータ：現在のパソコンなどのコンピュータの事を言う。

b) 振幅（確率振幅）とは

確率振幅の本体である**波動関数**はシュレーディンガーによって持ち込まれた。 この**波動関数**の絶対値の二乗を**確率**と考えるといろいろなものがうまく説明できるということにボルンという人が気づいた。二乗が**確率**なので一乗である**波動関数**は二乗して**確率**になるものという意味で**確率振幅**と呼ぶ。 （量子論における「確率振幅」とは何ですか？ - どこからそのような概念が生まれた... - Yahoo!知恵袋）

c) 重ね合わせとは

量子力学において、**重ね合わせ**（かさねあわせ、英: superposition）は、量子の振る舞いを計算する際に、定常状態と呼ばれるシンプルな性質を持つ複数の**波動関数**を**重ね合わせ**たものとして書き表すことである。（重ね合わせ - Wikipedia） 量子ビット状態を表現する (qiskit.org) での定常状態は$|0\rangle$、$|1\rangle$と表記されている。

d) 状態ベクトルとは

量子ビットは**複素ベクトル**によって記述される。この**複素ベクトル**のことを**状態ベクトル**と呼ぶ。
出典： （IBM Quantumで学ぶ量子コンピュータ/湊雄一郎/比嘉恵一朗/永井隆太郎/加藤拓己）

1. 量子力学関連

e) ブラ-ケット記法を使用した内積、テンソル積の表現

ベクトル$|a\rangle$, $|b\rangle$ の内積を、$\langle a|b\rangle$ で表す。また、$|a\rangle\langle b|$ は行列を表す。Qiskitテキストではこれを「外積」と呼ばれるとしているが、英語の「outer product」の和訳であり、別の概念である「vector product」や「cross product」の和訳も外積とされているため、注意が必要である。

f)量子力学の計算で出てくる記号の読み方

記号	黙読・読み方
λ	ラムダ
π	パイ
θ	シータ
$\otimes$	テンソル積
$\lvert\alpha\rvert^2$	アルファの絶対値の二乗
$\lvert\Phi\rangle$ $\lvert\phi\rangle$	ファイケット（大文字） ファイケット（小文字）

1. 量子力学関連

g) ブラ-ケット記法

物理学者ポール・ディラックが考案した記法。量子力学では量子状態（波動関数）を複素ベクトルで表すことが多く、このベクトルを縦1列のベクトルとして、記号　$|\Psi\rangle$ で表す。これをケットベクトルという。
また、ケットベクトルの各成分の共役複素数を成分に持つベクトルを転置した行ベクトルをブラベクトルといい、記号 $\langle\Psi|$ で表す。

記号	黙読・読み方	表現するもの	使用例
$\|\psi\rangle$	プサイケット ※数式中に小文字ψがある場合は、区別するために、「スモールプサイケット」と読むこともある。	縦1列のベクトル	$= \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \\ \vdots \end{pmatrix}$
$\|\Psi\rangle$	プサイケット ※数式中に大文字Ψがある場合は、区別するために、「ラージプサイケット」と読むこともある	縦1列のベクトル	$= \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \\ \vdots \end{pmatrix}$
$\langle\psi\|$	プサイブラ（スモールプサイブラ）	横1行のベクトル	$= (a_0^* \quad a_1^* \quad a_2^* \quad \cdots \quad a_n^* \quad \cdots)$
$\langle\Psi\|$	プサイブラ（ラージプサイブラ）	横1行のベクトル	$= (a_0^* \quad a_1^* \quad a_2^* \quad \cdots \quad a_n^* \quad \cdots)$
$\|\psi\rangle^\dagger$	プサイケットダガー	$\|\psi\rangle$の転置行列の各成分の複素共役を取ったもの。この二つのベクトルの関係を「随伴」あるいは「エルミート共役」と呼ぶ	$\|\psi\rangle^\dagger=\langle\psi\|$
ψ^*	プサイの複素共役	複素共役	

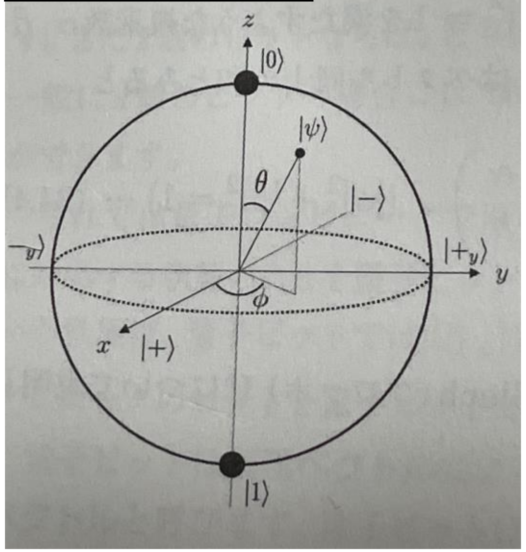
1. 量子力学関連

h) 位相とは

|0⟩と|1⟩の重ね合わせの量子状態を表す式

$$|\psi\rangle = e^{i\lambda}\left(\cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle\right)$$

ブロッホ球



①位相

ケットやブラに$\exp^{i\theta}$ $|\psi\rangle$のようにかかる複素数の係数（$\exp^{i\theta}$の部分）を位相と呼ぶ

②グローバル位相

状態ベクトル全体にかかる位相のことで、「|0⟩と|1⟩の重ね合わせの量子状態を表す式」では$e^{i\lambda}$です。グローバル位相には物理的な意味が無く、$|0\rangle + |1\rangle$と$e^{i\lambda}$（ $|0\rangle + |1\rangle$ ）は同じ状態となるので、グローバル位相はしばしば省略されます。

③相対位相

重ね合わせた2つの量子の相対的な位相差は物理的な意味を持ち、これを相対位相と呼ぶ。例えば$|0\rangle + |1\rangle$と$|0\rangle + e^{i\varphi}|1\rangle$は別の量子状態になります。

出典：（IBM Quantumで学ぶ量子コンピュータ/湊雄一郎/比嘉恵一朗/永井隆太郎/加藤拓己）

1. 量子力学関連

i) 複素振幅（英文：complex amplitude）（複素確率振幅）とは

量子力学では異なる2つの状態の**重ね合わせ**状態というのが許されているので、量子の世界の情報の最小単位である**量子ビット**は αとβという2つの**複素数**を用いた**複素ベクトル**を用いて $\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$ のようにその量子状態が記述される。
αやβはどの程度の重みで0状態と1状態が**重ね合わ**さっているかを表しており、**複素確率振幅**と呼ばれる。
αやβが**複素数**になっているのは、量子の世界では0や1といった離散的な量も波の性質を持ち干渉するためである。
古典ビットの0に対応する状態は $\begin{pmatrix}1\\ 0\end{pmatrix}$ 、1に対応する状態は $\begin{pmatrix}0\\ 1\end{pmatrix}$ となる。

j) 通信不可能定理

量子もつれ状態にある量子ビットの片方(A)を測定すると、もう片方の量子ビット(B)の状態も同時に決まるが、もう片方の量子ビット(B)の観測者には(A)の測定結果が伝わっていないため瞬時には(B)の状態は決まらない。
(B)の量子状態が瞬時に決まるということは光より速い速度で(A)の観測結果が(B)の観測者に伝わることを意味するため、ありえない。という定理。量子テレポーテーションのところで出てくる。
実際には、(A)の観測者から(B)の観測者へ測定結果を古典的手法で送らないといけないので光の速度より遅くなる。

1. 量子力学関連

k) 波動関数を複素数で表現する理由とは

量子状態を表現する**ベクトル**に**複素数**が出てくる理由は、**波動関数**が**複素数**で表現されているからである。
では、**波動関数**を**複素数**で表現する理由ですが、ディラックの教科書では次のような説明がされている。

量子力学では、**波動関数**の**振幅**を変えても状態としては同じものと見なされる。そうすると、2つの状態$|\mathrm{A}\rangle$と$|\mathrm{B}\rangle$を**重ね合わせ**た状態$|R\rangle = \mathrm{c1}|\mathrm{A}\rangle + \mathrm{c2}|\mathrm{B}\rangle$ (確率の和が1になるように、規格化条件$|c1|^2 + |c2|^2 =$

2. 数学関連

a) 複素数とは

実数部分と虚数部分の2つの要素からなる数（複数の要素からなる数）
○**複素数**の**ベクトル**での表現例

$$|q_0\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{bmatrix}$$

※ $|q_0\rangle$（読み方：キューゼロケット）
複素数a+ibをベクトルで(a, ib)と表す

b) ベクトルとは

大きさと向きを持つ量。

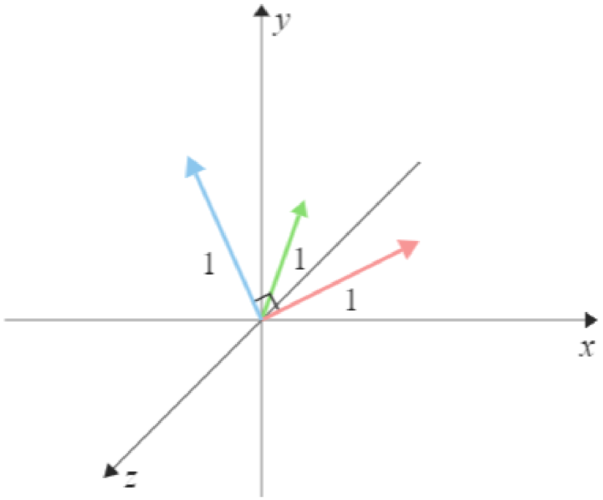
c) スカラーとは

特定の座標系とは無関係である量。
スカラーの数を**ベクトル**の**次元**（Dimension）と言う。
ベクトルには縦ベクトルと横ベクトルの2種類あり、**行列**（Matrix）を考える際に重要になる。

2. 数学関連

d) 正規直交基底とは

例えば、以下の赤、青、緑の3つの**ベクトル**は、それぞれ**単位ベクトル**であり、3次元空間の**正規直交基底**の一つである。



上記は、**直交基底**なので、3つの**ベクトル**はお互いに直交しています。この場合、3次元なので、青と赤の**ベクトル**、青と緑の**ベクトル**、緑と赤の**ベクトル**は、それぞれ90°の関係である。

直交基底となるそれぞれの**ベクトル**が**単位ベクトル**である**直交基底**のことを**正規直交基底**と言う。([正規直交基底とは - Cognicull](#))

2. 数学関連

e) オイラーの公式

$e^{i\pi} = \cos\pi + i\sin\pi = -1 + i \cdot 0 = -1$

f) マクローリン級数

$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} x^n$のべき級数

g) 2を法とする加算

加算の結果を2で割った余りを求める計算。例えば、0+0=0, 0+1=1, 1+1=2なので、
0⊕0=0, 0⊕1=1, 1⊕1=0となる。これは排他的論理和(XOR)と同じである。
※「法」の読み方は「ほう」と読む

3. Qiskit関連

a) 量子コンピュータ実機（IBM Quantum Systems）での実行と準備

①初めて実機（IBM Quantum Systems）で実行する場合に使うPython3コード

```
from qiskit import IBMQ
IBMQ.save_account('MY_API_TOKEN')    # MY_API_TOKENにご自分のトークンを入れてください
```

②２回目以降でMY_API_TOKEN変更時に実機（IBM Quantum Systems）で実行する場合に使うPython3コード

```
from qiskit import IBMQ
IBMQ.save_account('MY_API_TOKEN',overwrite=True)
```

としないとoverwrite=Trueがないと警告され実機が動作しない。

【参考】MY_API_TOKENの入手方法
【参考】最も空いている量子コンピュータ実機を教えて貰う為のPython3実装

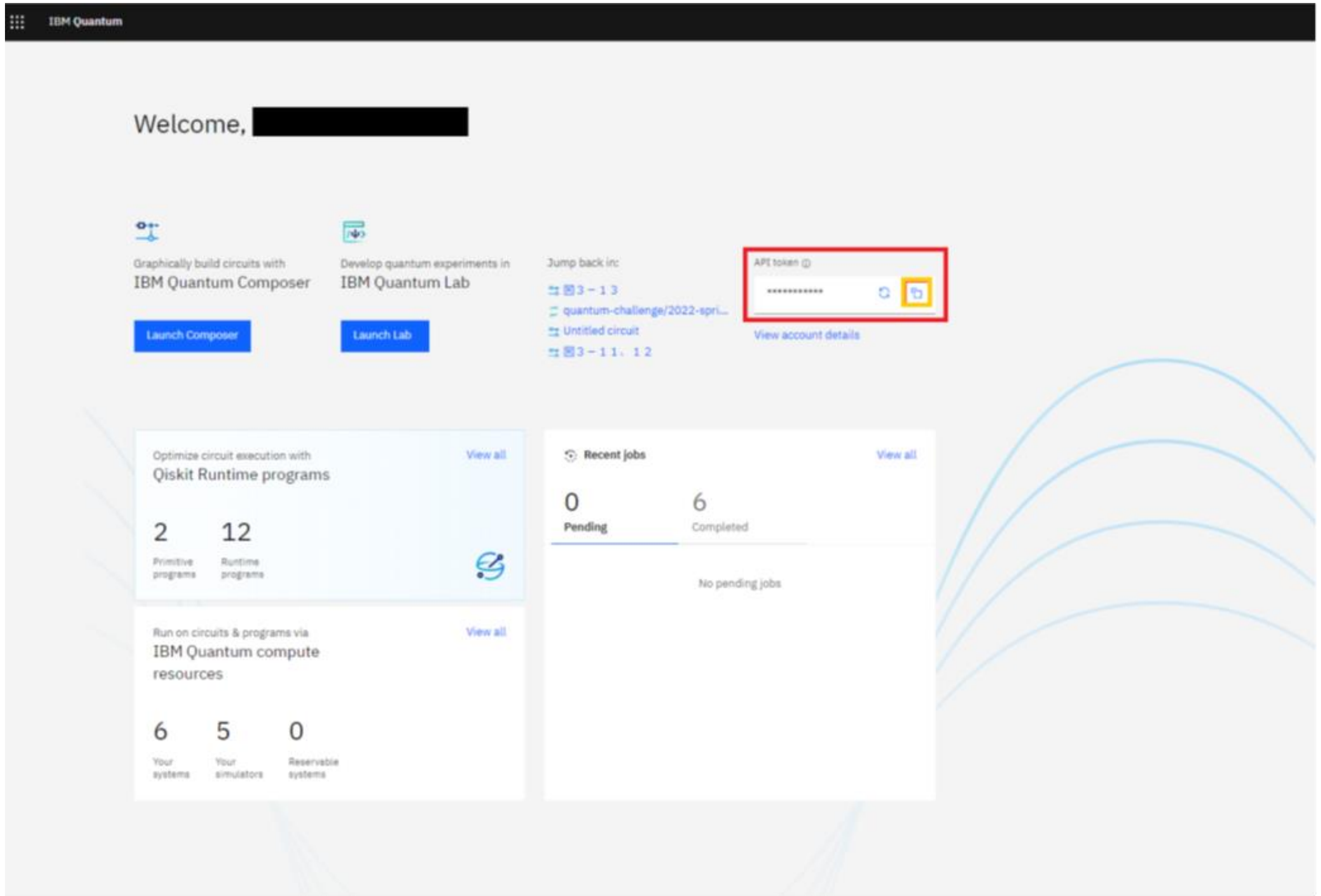
③実機で実行させる場合のPythonコード

```
provider=IBMQ.load_account()
backend=provider.get_backend('実機名')
```

3．Qiskit関連

【参考１】MY_API_TOKENの入手方法

IBM Quantumへログインして表示される画面の右上にAPI token（赤枠）と言う欄がある。このコピーボタン（オレンジ枠）をクリックすると自分のAPI tokenがクリップボードへコピーされる。
これをJupyter Notebook(ジュパイターノートブックまたはジュピターノートブックと呼ぶ）に貼り付け利用する。



3. Qiskit関連

【参考２】最も空いている量子コンピュータ実機を教えて貰う為のPython3実装

```
# initialization
import matplotlib.pyplot as plt
import numpy as np

# importing Qiskit
from qiskit import IBMQ, Aer
from qiskit.providers.ibmq import least_busy
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister, transpile, assemble

# import basic plot tools
from qiskit.visualization import plot_histogram


# Load our saved IBMQ accounts and get the least busy backend device with less than or equal to 5 qubits
IBMQ.load_account()
provider = IBMQ.get_provider(hub='ibm-q')
provider.backends()
backend = least_busy(provider.backends(filters=lambda x: x.configuration().n_qubits <= 5 and
                              x.configuration().n_qubits >= 2 and
                              not x.configuration().simulator and x.status().operational==True))
print("最も空いている実機: ", backend)
```

実行結果：最も空いている実機: ibmq_belem

3. Qiskit関連

b) 量子コンピュータシミュレータでの実行

①シミュレータで実行させる場合のPython3コード

backend=qiskit.Aer.get_backend('シミュレータ名')

シミュレータと実装例
'statevector_simulator'：理想的な量子回路における出力状態ベクトルを計算するシミュレータ
'unitary_simulator'：理想的な量子回路と等価なユニタリ変換行列を計算するシミュレータ
'qasm_simulator'：複数回の測定・サンプリングを伴う量子回路実行が可能なシミュレータ
例：backend=qiskit.Aer.get_backend('qasm_simulator')